

Critical Database Size for Effective Caching

Ajaykrishnan N., Navya S. Prem

Dept. of Electronics and Communication Engineering

National Institute of Technology Karnataka

{aja.11ec09, nsp.11ec51}@nitk.edu.in

Vinod M. Prabhakaran, Rahul Vaze

School of Technology and Computer Science

Tata Institute of Fundamental Research, Mumbai

{vinodmp, vaze}@tifr.res.in

Abstract

Replicating or caching popular content in memories distributed across the network is a technique to reduce peak network loads. Conventionally, the performance gain of caching was thought to result from making part of the requested data available closer to end users. Recently, it has been shown that by using a carefully designed technique to store the contents in the cache and coding across data streams a much more significant gain can be achieved in reducing the network load. Inner and outer bounds on the network load v/s cache memory tradeoff were obtained in [1]. We give an improved outer bound on the network load v/s cache memory tradeoff. We address the question of to what extent caching is effective in reducing the server load when the number of files becomes large as compared to the number of users. We show that the effectiveness of caching become small when the number of files becomes comparable to the square of the number of users.

I. INTRODUCTION

In recent times, there has been an increase in demand for online video streaming leading to high data traffic. Also, it is observed that the demands are variable across time, with periods of high and low traffic demand. The load on the server is high during peak hours when a majority of users access video and relatively low at other times. Thus, there exists the possibility of storing content at the end users during the off peak hours such that the load on the server is reduced during peak hours. This method is called *caching*. There are two main phases involved in this process, placement phase and delivery phase. In the placement phase, data is stored at the end user when the network is relatively uncongested; here the constraint is the cache memory size at the user. Also, at this stage the actual request the user might make is not usually known. In the delivery phase, when the actual requests of the users are made, the constraint is the rate required to serve all the requested content.

A straightforward approach is to cache a copy of a fraction of all the files at all the users. Then in the delivery phase, the central server needs to send only the remaining parts of the requested files. This is effective only when the cache size is comparable to the database size at the server.

A more sophisticated approach is to allow the central server to satisfy the request of several users with different demands with a single multicast stream as was shown in [1] using the idea of network coding [2]. Streams are generated by coding across the different files requested. This reduces the rate as compared to a conventional caching scheme. The requested files are decoded from the data stream using the contents stored in the local cache memory. The gain from this approach is not only proportional to the cache size but also increases with the increasing number of users. Another approach suggested in [1] is to store contents that are coded across files to reduce the rate.

In [1], inner and outer bounds on the optimal tradeoff between cache size M at each user and the data rate R required to service any set of single file requests from all the users were obtained. Considering a popularity distribution on the files, inner and outer bounds on the tradeoff between cache size and expected load of the shared link was obtained in [3]. An online version of this problem was considered in [4]. In [5], a scheme was proposed where the placement phase is distributed and not centrally controlled by the central server. In [6], a hierarchical system is considered, where caching happens at two or more levels.

In this paper, we are interested in the case when the database size is large compared to the number of users. For a fixed cache size, when the number of files is considerably large compared to the number of users, no significant gain in the rate can be achieved by any scheme compared to having no cache. Specifically, we are interested in finding the minimum number of files beyond which the benefits of caching disappear in the setting of [1]. To this end, we first prove a general outer bound on the optimal (M, R) tradeoff which generalizes an example in [1]. We show that the gains from caching are small when the number of files is comparable to the square of the number of users. We

then define the pre-constant to the $\Theta(K^2)$ term (where K denotes the number of users). Using the improved outer bound we obtain a better upper bound to this pre-constant.

The rest of the paper is organized as follows. In Section II we recapitulate the system model proposed in [1], and in Section III we summarize the different caching strategies proposed there. We derive a new outer bound on the tradeoff of cache size and rate in Section IV by generalizing an example in [1]. In Section V, we calculate the minimum number of files beyond which benefits of caching become small. We finish with a short discussion in Section VI.

II. SYSTEM MODEL

Consider a system (see Fig. 1) with K users connected to the central server through a shared, error free link. The server has access to the database containing N files W_1, \dots, W_N , of F bits each, all independent and uniformly distributed. Each user has access to a cache Z_k of size MF bits for some real number $M \in [0, N]$. In the placement

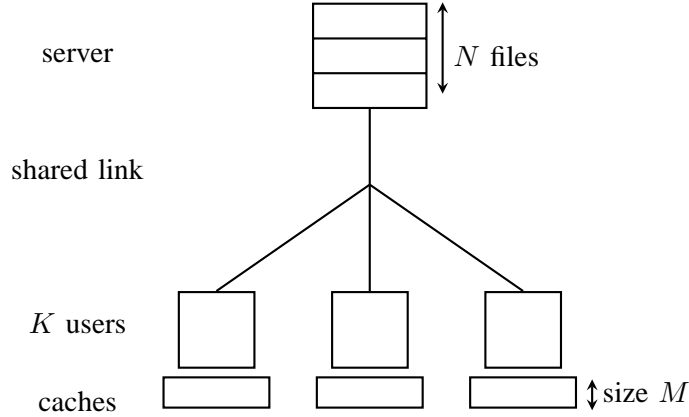


Fig. 1. Caching system consisting of N files at the server, K users each having a cache of size M files as in [1].

phase, the user fills the content of its cache by accessing the database. In the delivery phase, user k requests one of the files W_{d_k} from the database. The server knows all the requests and transmits a signal $X_{(d_1, \dots, d_K)}$ of size at most RF bits, where we call R the rate and (d_1, \dots, d_K) the file request vector. Using the content Z_k of its cache and the signal received $X_{(d_1, \dots, d_K)}$, each user k must decode its requested file W_{d_k} . For the rest of the paper we will be expressing R and M as well as entropies and mutual informations in units of F bits.

Definition 1. The memory-rate pair (M, R) is *achievable* if for every $\varepsilon > 0$ and every large enough file size F there exists an (M, R) caching scheme such that the probability of error in decoding the required file is less than ε for each request vector. We define the optimal *memory-rate tradeoff* as

$$R^*(M) \triangleq \inf\{R : (M, R) \text{ is achievable}\}.$$

III. CACHING STRATEGIES

We summarize the three strategies given in [1]. Here, coding refers to taking linear combinations of the requested files.

A. Uncoded Caching

There is no coding involved in this strategy. Each user caches $\frac{M}{N}$ fraction of each file in the placement phase and in the delivery phase the $1 - \frac{M}{N}$ fraction of the file that is not available to the user is transmitted by the server. Since there are N files, and the size of each file is F bits, the cache size of each user is MF bits. In the worst case, when no two users request the same file, for each of the K users, the server needs to transmit the remaining $1 - \frac{M}{N}$ part of each file. This gives an achievable rate $R_U(M)$ which is,

$$R_U(M) \triangleq K \left(1 - \frac{M}{N}\right) \cdot \min \left\{1, \frac{N}{K}\right\}. \quad (1)$$

There are two factors, K which is the rate without caching and $1 - \frac{M}{N}$, which is the gain because of the availability of caches at the end user referred to as *local caching gain*. When the number of users is more than the number of files then an additional gain of $\frac{N}{K}$ is obtained.

B. Coded Caching

In this strategy, as mentioned before, the aim is to multicast (combine various files meant for different users) in the delivery phase. In the placement phase, each file is divided into $\left(\frac{K}{N}\right)$ equal-sized parts, and each user caches $\frac{MF}{N}$ bits of each file such that every $\frac{MK}{N}$ set of users have one part of each file in common. For the delivery phase, consider any set of $\frac{MK}{N} + 1$ users. Each user in the set will require a part of the requested file that is present at the remaining $\frac{MK}{N}$ users in the set. The central server sends a linear combination of all the $\frac{MK}{N} + 1$ requested parts. Similar linear combinations are sent by considering all possible sets of $\frac{MK}{N} + 1$ users. This gives an achievable rate $R_C(M)$ of [1],

$$R_C(M) \triangleq K \cdot \left(1 - \frac{M}{N}\right) \cdot \min \left\{ \frac{1}{1 + \frac{KM}{N}}, \frac{N}{K} \right\}. \quad (2)$$

In addition to the *local caching gain* as explained in section III-A, coded caching achieves an additional gain of $\frac{1}{1 + \frac{KM}{N}}$ which is the *global caching gain*.

C. Coded Content Placement

The achievable rate of section III-B can be further improved by coded content placement. For $M = \frac{1}{N}$, coded content placement strategy has a lower rate compared to coded caching strategy which improves the rate in the region $M = (0, 1)$. We illustrate this with an example. Consider the case of $N = K = 3$ and $M = 1/3$. In this strategy, we split the three files A, B, C into three sub files i.e., $A = (A_1, A_2, A_3)$, $B = (B_1, B_2, B_3)$ and $C = (C_1, C_2, C_3)$. The caches are stored with $Z_1 = A_1 \oplus B_1 \oplus C_1$, $Z_2 = A_2 \oplus B_2 \oplus C_2$ and $Z_3 = A_3 \oplus B_3 \oplus C_3$. Consider that user one requests file A, user two requests file B and user three request file C. The server satisfies the requests by transmitting $(B_1, C_1, A_2, C_2, A_3, B_3)$ at rate $R = 2$ which does better than the achievable rate $R_C(M)$ given by (2) as shown in Fig. 2.

IV. LOWER BOUND ON $R^*(M)$

In this section, we first summarize the cut-set bound of [1] and then give an improved bound.

A. Cut-Set Bound

Let $s \in \{1, \dots, \min\{N, K\}\}$. Consider $X_{(1,2,\dots,s)}$, which is transmitted during the delivery phase, on the shared link when the first s users request files $1, 2, \dots, s$, respectively. Then, $X_{(1,2,\dots,s)}$ along with the caches Z_1, \dots, Z_s of the first s users must determine the files W_1, \dots, W_s . In a similar manner consider $X_{(s+1,\dots,2s)}, \dots, X_{((\lfloor N/s \rfloor - 1)s + 1, \dots, (\lfloor N/s \rfloor)s)}$. Now $X_{(1,2,\dots,s)}, \dots, X_{((\lfloor N/s \rfloor - 1)s + 1, \dots, (\lfloor N/s \rfloor)s)}$ and Z_1, \dots, Z_s must determine $W_1, \dots, W_{\lfloor N/s \rfloor}$. Since $\lfloor N/s \rfloor$ transmissions of size R and s caches of size M determines $s \lfloor N/s \rfloor$ files we have,

$$\lfloor N/s \rfloor R^*(M) + sM \geq s \lfloor N/s \rfloor.$$

Solving for $R^*(M)$ and optimizing over all s , we obtain

$$R^*(M) \geq \max_{s \in \{1, \dots, \min\{N, K\}\}} \left(s - \frac{s}{\lfloor N/s \rfloor} M \right). \quad (3)$$

B. An Improved Bound - An Example

In this section, we give an example to illustrate how the lower bound on $R^*(M)$ can be tightened compared to the cut-set bound (3) by generalizing the approach used in [1, Appendix].

Example 1. Consider the case of $N = 9$ files and $K = 4$ users. We consider $X_{1245}, X_{3167}, X_{8912}$ and X_{7431} , the signals transmitted by the server for the request vectors $(1, 2, 4, 5), (3, 1, 6, 7), (8, 9, 1, 2)$ and $(7, 4, 3, 1)$, respectively. W_1 can be decoded by user 1 using its cache Z_1 and X_{1245} . Similarly, user 2 can decode file W_1 using Z_2 and X_{3167} . In the same way, users 3 and 4 can decode file W_1 from their caches along with X_{8912} and X_{7431} , respectively. Now, notice that W_2 and W_3 can be decoded by combining X_{1245}, X_{3167} and the caches Z_1 of user 1 and Z_2 of user 2. Specifically, user 1 with its cache Z_1 and X_{3167} can decode file W_3 and user 2 with its cache Z_2 and X_{1245} can decode file W_2 . In the same way, files W_2 and W_3 can also be decoded by combining X_{8912}, X_{7431} and the caches Z_3 of user 3 and Z_4 of user 4. This combining refers to step (b) in the chain of inequalities below and is key to obtaining our lower bound. The remaining files $(W_4, W_5, W_6, W_7, W_8, W_9)$ can be decoded by taking all the 4 request vectors together and using the corresponding cache of the user that requests that file. The steps given below demonstrates this procedure. Recall that R, M , entropies, and mutual informations are all in units of F bits. For any achievable memory-rate pair (M, R) , (below we suppress the small terms resulting from Fano's inequality)

$$\begin{aligned}
4M + 4R &\geq H(X_{1245}, Z_1) + H(X_{3167}, Z_2) + H(X_{8912}, Z_3) + H(X_{7431}, Z_4) \\
&= H(X_{1245}, Z_1|W_1) + I(W_1; X_{1245}, Z_1) + H(X_{3167}, Z_2|W_1) + I(W_1; X_{3167}, Z_2) + \\
&\quad H(X_{8912}, Z_3|W_1) + I(W_1; X_{8912}, Z_3) + H(X_{7431}, Z_4|W_1) + I(W_1; X_{7431}, Z_4) \\
&\stackrel{(a)}{\geq} H(X_{1245}, Z_1|W_1) + H(X_{3167}, Z_2|W_1) + H(X_{8912}, Z_3|W_1) + H(X_{7431}, Z_4|W_1) + 4 \\
&\stackrel{(b)}{\geq} H(X_{1245}, Z_1, X_{3167}, Z_2|W_1) + H(X_{8912}, Z_3, X_{7431}, Z_4|W_1) + 4 \\
&= H(X_{1245}, Z_1, X_{3167}, Z_2|W_1, W_2, W_3) + I(W_2, W_3; X_{1245}, Z_1, X_{3167}, Z_2|W_1) + \\
&\quad H(X_{8912}, Z_3, X_{7431}, Z_4|W_1, W_2, W_3) + I(W_2, W_3; X_{8912}, Z_3, X_{7431}, Z_4|W_1) + 4 \\
&\geq H\left(\begin{array}{c} X_{1245}, Z_1, X_{3167}, Z_2, \\ X_{8912}, Z_3, X_{7431}, Z_4 \end{array} \middle| \begin{array}{c} W_1, \\ W_2, W_3 \end{array}\right) + I(W_2, W_3; X_{1245}, Z_1, X_{3167}, Z_2|W_1) + \\
&\quad I(W_2, W_3; X_{8912}, Z_3, X_{7431}, Z_4|W_1) + 4 \\
&\stackrel{(c)}{\geq} I\left(\begin{array}{c} W_4, W_5, W_6, W_7, W_8, W_9; \\ X_{1245}, Z_1, X_{3167}, Z_2, \\ X_{8912}, Z_3, X_{7431}, Z_4 \end{array} \middle| \begin{array}{c} W_1, \\ W_2, \\ W_3 \end{array}\right) + 8 \\
&\stackrel{(d)}{=} 14,
\end{aligned}$$

where (a) follows from Fano's inequality since W_1 can be decoded from each of $(X_{1245}, Z_1), (X_{3167}, Z_2), (X_{8912}, Z_3)$ and (X_{7431}, Z_4) , and (b) holds because

$$\begin{aligned}
H(X_{1245}, Z_1|W_1) + H(X_{3167}, Z_2|W_1) &\geq H(X_{1245}, Z_1, X_{3167}, Z_2|W_1), \\
H(X_{8912}, Z_3|W_1) + H(X_{7431}, Z_4|W_1) &\geq H(X_{8912}, Z_3, X_{7431}, Z_4|W_1).
\end{aligned}$$

Similarly (c) follows from Fano's inequality because W_2, W_3 can be decoded from each of $(X_{1245}, Z_1, X_{3167}, Z_2)$ and $(X_{8912}, Z_3, X_{7431}, Z_4)$. Similarly, (d) holds because $(W_4, W_5, W_6, W_7, W_8, W_9)$ can be decoded from $(X_{1245}, Z_1, X_{3167}, Z_2, X_{8912}, Z_3, X_{7431}, Z_4)$. Combining the above results we get,

$$M + R^*(M) \geq 3.5.$$

This is an improvement over the cut-set bound which gives $M + R^*(M) \geq 3$. The coded caching achievable strategy gives $\inf_{M \geq 0} M + R_C(M) = 3.75$ at $M = 2.25$. \square

C. General Lower Bound

Our main result is the following lower bound on the optimal (M, R) tradeoff. Recall that M, R are in units of F bits.

Theorem 1.

For $\alpha > 0$ and $K \geq 2$ users, if (M, R) is achievable,

(i) then for $N \geq \lceil \frac{1}{\alpha} \rceil$,

$$\alpha M + R \geq \begin{cases} \frac{N - \lceil \frac{1}{\alpha} \rceil ((n - \gamma)^2 - (n - \gamma) + 1)}{2 \lceil \frac{1}{\alpha} \rceil (n - \gamma)} + (n - \gamma), & N \leq \lceil \frac{1}{\alpha} \rceil (3(n - \gamma)^2 - (n - \gamma) + 1) \\ 2(n - \gamma), & N > \lceil \frac{1}{\alpha} \rceil (3(n - \gamma)^2 - (n - \gamma) + 1) \end{cases} \quad (4)$$

where,

$$n = \left\lceil \frac{\lceil \frac{1}{\alpha} \rceil + \sqrt{\lceil \frac{1}{\alpha} \rceil^2 + 12 \lceil \frac{1}{\alpha} \rceil (N - \lceil \frac{1}{\alpha} \rceil)}}{6 \lceil \frac{1}{\alpha} \rceil} \right\rceil, \quad (5)$$

$$\gamma = \max \left(0, \left\lceil n - \frac{K}{2} \right\rceil \right). \quad (6)$$

(ii) then for $N < \lceil \frac{1}{\alpha} \rceil$,

$$\alpha M + R \geq \frac{N}{\lceil \frac{1}{\alpha} \rceil}. \quad (7)$$

For $\alpha > 1$ and $K \geq 2 \lfloor \alpha \rfloor$ users, if (M, R) is achievable,

(i) then for $N \geq \lfloor \alpha \rfloor$,

$$\alpha M + R \geq \begin{cases} \frac{N - \lfloor \alpha \rfloor ((n - \gamma)^2 - (n - \gamma) + 1)}{2(n - \gamma)} + (n - \gamma) \lfloor \alpha \rfloor, & N \leq \lfloor \alpha \rfloor (3(n - \gamma)^2 - (n - \gamma) + 1) \\ 2(n - \gamma) \lfloor \alpha \rfloor, & N > \lfloor \alpha \rfloor (3(n - \gamma)^2 - (n - \gamma) + 1) \end{cases} \quad (8)$$

where,

$$n = \left\lceil \frac{\lfloor \alpha \rfloor + \sqrt{\lfloor \alpha \rfloor^2 + 12 \lfloor \alpha \rfloor (N - \lfloor \alpha \rfloor)}}{6 \lfloor \alpha \rfloor} \right\rceil, \quad (9)$$

$$\gamma = \max \left(0, \left\lceil n - \frac{K}{2 \lfloor \alpha \rfloor} \right\rceil \right). \quad (10)$$

(ii) then for $N < \lfloor \alpha \rfloor$,

$$\alpha M + R \geq N. \quad (11)$$

A proof is given in the Appendix. The next example also shows that, in general, Theorem 1 is tighter than the cut-set bound (3).

Example 2. Consider the case of $N = 3$ files and $K = 3$ users. The cut-set lower bound (3), the lower bound of (4) for $\alpha = 1$, and the achievable tradeoffs of (1) and (2) are shown in Figure 2.

V. CRITICAL DATABASE SIZE FOR EFFECTIVE CACHING

For any caching system, if the number of files grows we expect the reduction in R to be small, for a fixed number of users K and cache size M . In general, each user may find only a small fraction of the file requested in its cache. This results in the server having to send a significant part of the requested file in most cases. So the decrease in rate R for a fixed M is negligible. Hence, having a large database decreases the benefits of caching.

To find the minimum database size for a fixed number of users for which caching becomes ineffective, we consider the quantity $(\alpha M + R^*(M))$, which arguably measures the cost of operating a caching system, where $\alpha > 0$ is the relative cost of cache memory (per user) versus server bandwidth. Clearly,

$$\inf_{M \geq 0} (\alpha M + R^*(M)) \leq K,$$

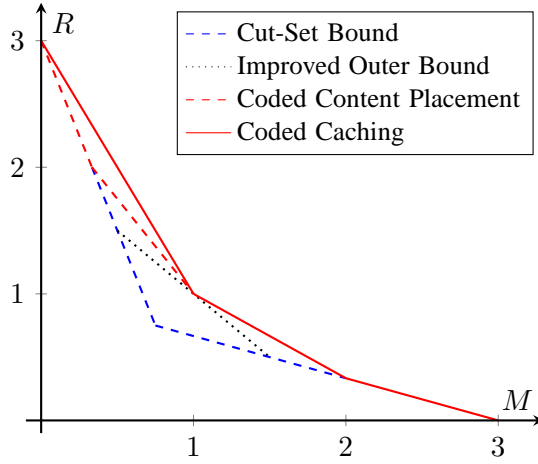


Fig. 2. The (M, R) tradeoff for $N = 3$ files and $K = 3$ users.

since $R^*(M) = K$ for $M = 0$, as the central server must serve the whole file when there is no cache. We are interested in finding the smallest size of the database, such that $\inf_{M \geq 0} (\alpha M + R^*(M)) = K$.

Definition 2. For any K users and $\alpha > 0$, $N(\alpha, K)$ is the minimum number of files such that

$$\inf_{M \geq 0} (\alpha M + R^*(M)) = K.$$

The following three lemmas give upper and lower bounds on $N(\alpha, K)$. Lemma 1 uses the cut set bound to derive an upper bound on $N(\alpha, K)$. An improved upper bound using Theorem 1 is given in Lemma 2. A lower bound on $N(\alpha, K)$ using the coded caching achievable strategy of [1] is given in Lemma 3.

Lemma 1. For K users and $\alpha > 0$,

$$N(\alpha, K) \leq \left\lceil \frac{1}{\alpha} \right\rceil K^2.$$

Using the lower bound we derived in Theorem 1, we can improve upon this bound. We illustrate this with an example.

Example 3. Consider the case when there are $K = 4$ users and instead of 9 files considered in Section IV-B, suppose we increase the number of files to $N = 11$. Following the same procedure as in Example 1, we get

$$M + R^*(M) \geq 4.$$

Thus upper bound on

$$N(\alpha, K) = 11.$$

This is an improvement compared to $N = 16$ files given by Lemma 1. □

Lemma 2. For $K \geq 2$ users and $\alpha > 0$,

$$N(\alpha, K) \leq \left\lceil \frac{1}{\alpha} \right\rceil \left(3 \left\lceil \frac{K}{2} \right\rceil^2 - \left\lceil \frac{K}{2} \right\rceil + 1 \right).$$

For $K \geq 2 \lfloor \alpha \rfloor$ users and $\alpha > 1$,

$$N(\alpha, K) \leq \lfloor \alpha \rfloor \left(3 \left\lceil \frac{K}{2 \lfloor \alpha \rfloor} \right\rceil^2 - \left\lceil \frac{K}{2 \lfloor \alpha \rfloor} \right\rceil + 1 \right).$$

Lemma 3. For K users and $\alpha > 0$,

$$N(\alpha, K) \geq \frac{1}{\alpha} \left(\frac{K^2}{2} + \frac{K}{2} \right).$$

The proofs of the lemmas are given in the Appendix. From the lemmas it is clear that $N(\alpha, K) = \Theta(K^2)$. Thus, it is important to characterize the smallest pre-constant to the $\Theta(K^2)$ term which is concretely defined as,

$$\beta_\alpha \triangleq \lim_{K \rightarrow \infty} \frac{N(\alpha, K)}{K^2}.$$

The following theorem directly follows from the lemmas.

Theorem 2. For any K users, $\alpha > 0$ and $N(\alpha, K)$, β_α is bounded by

$$\begin{cases} \left(\frac{1}{\alpha}\right)^{\frac{1}{2}} \leq \beta_\alpha \leq \left\lceil \frac{1}{\alpha} \right\rceil^{\frac{3}{4}} & , 0 < \alpha \leq 1 \\ \left(\frac{1}{\alpha}\right)^{\frac{1}{2}} \leq \beta_\alpha \leq \frac{1}{\lfloor \alpha \rfloor^{\frac{3}{4}}} & , \alpha > 1. \end{cases}$$

Since the minimum number of files $N(\alpha, K)$ such that $\inf_{M \geq 0} (\alpha M + R^*(M)) = K$ is of $\Theta(K^2)$, we can conclude that the effectiveness of caching becomes small when the number of files becomes comparable to the square of the number of users.

VI. DISCUSSION

In this paper, we consider the case when the number of files is large compared to the number of users in a caching system. First, we studied inner and outer bounds on the memory-rate tradeoff and present an improved outer bound by generalizing the approach used in [1]. We showed that when the number of files is comparable to the square of the number of users, the benefits of caching become negligible. We defined the β_α to be the pre-constant to the $\Theta(K^2)$ term. Using the improved bound, we obtain a better upper bound to this pre-constant.

We studied the worst-case shared link load (as in [1]). We expect similar results to hold for the expected load of the shared link under popularity distributions on files with a large number of popular files.

ACKNOWLEDGEMENT

This work was supported in part by Information Technology Research Academy (ITRA), Government of India under ITRA-Mobile grant ITRA/15(64)/Mobile/USEAADWN/01.

REFERENCES

- [1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, Apr. 2000.
- [3] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," in *Proc. of IEEE INFOCOM WORKSHOPS*, pp. 221 - 226, May 2014.
- [4] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," in *Proc. of IEEE ICC*, pp. 1878 - 1883, June 2014.
- [5] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," in *Proc. of 51st Annual Allerton Conference on Communication, Control, and Computing*, pp. 421 - 427, Oct. 2013.
- [6] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. Diggavi, "Hierarchical Coded Caching," in *Proc. of IEEE ISIT*, pp. 2142 - 2146, July 2014.

APPENDIX

PROOF OF THEOREM 1

We will first obtain a lower bound on $(M + R)$, for any achievable (M, R) , i.e., the case of $\alpha = 1$. For this, we first consider the case of $K \geq 2n$, where n is as defined in (5). Note that γ of (6) is 0 in this case. Recall Example

1 where 4 request vectors were considered. Similarly, we consider the following $2n$ request vectors

$$(1, u_1^1, \dots, u_{n-1}^1, v_1^1, \dots, v_n^1, t_1^1, \dots, t_{K-2n}^1) \quad (12a)$$

$$(u_1^2, 1, \dots, u_{n-1}^2, v_1^2, \dots, v_n^2, t_1^2, \dots, t_{K-2n}^2) \quad (12b)$$

\vdots

$$(u_1^n, \dots, u_{n-1}^n, 1, v_1^n, \dots, v_n^n, t_1^n, \dots, t_{K-2n}^n) \quad (12c)$$

$$(v_1^{n+1}, \dots, v_n^{n+1}, 1, u_1^1, \dots, u_{n-1}^1, t_1^{n+1}, \dots, t_{K-2n}^{n+1}) \quad (12d)$$

\vdots

$$(v_1^{2n}, \dots, v_n^{2n}, u_1^n, \dots, u_{n-1}^n, 1, t_1^{2n}, \dots, t_{K-2n}^{2n}) \quad (12e)$$

Of these, we require that $1, u_1^1, \dots, u_{n-1}^1, \dots, u_1^n, \dots, u_{n-1}^n$ be distinct. Hence, we will require that $n^2 - n + 1 \leq N$. Furthermore, we want these along with the v 's, i.e., $1, u_1^1, \dots, u_{n-1}^1, \dots, u_1^n, \dots, u_{n-1}^n, v_1^1, \dots, v_n^1, \dots, v_1^{2n}, \dots, v_n^{2n}$ to include all of $1, 2, \dots, N$. Hence, we need n to be such that

$$n^2 - n + 1 \leq N \leq 3n^2 - n + 1. \quad (13)$$

We can verify that the choice of n in (5), which is reproduced below, satisfies this.

$$n = \left\lceil \frac{1 + \sqrt{1 + 12(N - 1)}}{6} \right\rceil.$$

Consider the first request vector and the first n users. User 1 requests file W_1 , and the rest $n - 1$ users request files $(W_{u_1^1}, \dots, W_{u_{n-1}^1})$. Similarly, in the second request vector, user 2 requests file W_1 and the rest $n - 1$ users request files $(W_{u_1^2}, \dots, W_{u_{n-1}^2})$. In the same manner for the n -th request vector, user n requests file W_1 and the first $n - 1$ users request files $(W_{u_1^n}, \dots, W_{u_{n-1}^n})$. These $(W_1, W_{u_1^1}, \dots, W_{u_{n-1}^n})$ are $n^2 - n + 1$ distinct files in the database. For the second set of n request vectors, users $n + 1$ to $2n$ request the same files as users 1 to n in the first n request vectors. For the first n request vectors, users $n + 1$ to $2n$ requests n^2 files $(W_{v_1^1}, \dots, W_{v_n^n})$. For the second n request vectors, users 1 to n requests n^2 files $(W_{v_1^{n+1}}, \dots, W_{v_n^{n+1}})$. By our choices we have ensured that these $2n^2$ files contain the remaining $N - (n^2 - n + 1)$ distinct files.

We now follow the same procedure as in Example 1. First file W_1 can be decoded from all the $2n$ request vectors. This is done by considering the first request vector and cache Z_1 , the second request vector and cache Z_2 and so on for the remaining request vectors. Then, the first set of n vectors and the second set of n vectors are separately combined to decode files $(W_{u_1^1}, \dots, W_{u_{n-1}^n})$. From the first n request vectors and caches (Z_1, \dots, Z_n) the files $(W_{u_1^1}, \dots, W_{u_{n-1}^n})$ can be decoded. Similarly, from the second set of n vectors and (Z_{n+1}, \dots, Z_{2n}) the same set of files can be decoded. The rest $N - (n^2 - n + 1)$ files which are included in $(W_{v_1^1}, \dots, W_{v_n^n})$ can be decoded by considering all the $2n$ request vectors together using all the caches (Z_1, \dots, Z_{2n}) . All R , M , entropies and mutual informations are in units of F bits and, as before, we suppress small terms from Fano's inequality. So for any achievable memory-rate pair (M, R) and $K \geq 2n$,

$$\begin{aligned} 2n(M + R) &\geq H(X_{(1, u_1^1, \dots, u_{n-1}^1, v_1^1, \dots, v_n^1, t_1^1, \dots, t_{K-2n}^1)}, Z_1) + \dots + \\ &\quad H(X_{(u_1^n, \dots, u_{n-1}^n, 1, v_1^n, \dots, v_n^n, t_1^n, \dots, t_{K-2n}^n)}, Z_n) + \\ &\quad H(X_{(v_1^{n+1}, \dots, v_n^{n+1}, 1, u_1^1, \dots, u_{n-1}^1, t_1^{n+1}, \dots, t_{K-2n}^{n+1})}, Z_{n+1}) \\ &\quad + \dots + H(X_{(v_1^{2n}, \dots, v_n^{2n}, u_1^n, \dots, u_{n-1}^n, 1, t_1^{2n}, \dots, t_{K-2n}^{2n})}, Z_{2n}) \\ &\stackrel{(i)}{\geq} H \left(\begin{array}{c} X_{(1, u_1^1, \dots, u_{n-1}^1, v_1^1, \dots, v_n^1, t_1^1, \dots, t_{K-2n}^1)}, \dots \\ X_{(u_1^n, \dots, u_{n-1}^n, 1, v_1^n, \dots, v_n^n, t_1^n, \dots, t_{K-2n}^n)}, \\ Z_1, \dots, Z_n \end{array} \middle| W_1 \right) + \end{aligned}$$

$$\begin{aligned}
& H \left(\begin{array}{c} X(v_1^{n+1}, \dots, v_n^{n+1}, 1, u_1^1, \dots, u_{n-1}^1, t_1^{n+1}, \dots, t_{K-2n}^{n+1}), \\ \dots, X(v_1^{2n}, \dots, v_n^{2n}, u_1^n, \dots, u_{n-1}^n, 1, t_1^{2n}, \dots, t_{K-2n}^{2n}), \\ Z_{n+1}, \dots, Z_{2n} \end{array} \middle| W_1 \right) \\
& + 2n \\
& \stackrel{(ii)}{\geq} H \left(\begin{array}{c} X(1, u_1^1, \dots, u_{n-1}^1, v_1^1, \dots, v_n^1, t_1^1, \dots, t_{K-2n}^1), \dots, \\ X(u_1^n, \dots, u_{n-1}^n, 1, v_1^n, \dots, v_n^n, t_1^n, \dots, t_{K-2n}^n) \\ X(v_1^{n+1}, \dots, v_n^{n+1}, 1, u_1^1, \dots, u_{n-1}^1, t_1^{n+1}, \dots, t_{K-2n}^{n+1}), \\ \dots, X(v_1^{2n}, \dots, v_n^{2n}, u_1^n, \dots, u_{n-1}^n, 1, t_1^{2n}, \dots, t_{K-2n}^{2n}) \\ Z_1, \dots, Z_{2n} \end{array} \middle| \mathbf{W} \right) \\
& + 2n + 2n(n-1) \\
& \stackrel{(iii)}{\geq} 2n^2 + (N - (n^2 - n + 1)),
\end{aligned}$$

where (i) is similar to steps (a) and (b) together in Example 1. In step (ii), which is similar to step (c) in Example 1. We define

$$\mathbf{W} = (W_1, W_{u_1^1}, \dots, W_{u_{n-1}^1}, \dots, W_{u_1^n}, \dots, W_{u_{n-1}^n}).$$

Step (iii) is similar to step (d) of Example 1. Therefore, for $K \geq 2n$,

$$M + R \geq n + \frac{N - (n^2 - n + 1)}{2n}. \quad (14)$$

Notice that $\gamma = 0$ for $K \geq 2n$, and the definition of n is such that $N \leq 3n^2 - n + 1$. Thus we have proved the theorem for $\alpha = 1$, $K \geq 2n$.

When $K < 2n$, we defined $\gamma \geq 0$ as the smallest integer such that $K \geq 2(n - \gamma)$. Notice that since $K \geq 2$, $(n - \gamma) > 0$. Recall that we had considered $2n$ vectors. Now we consider $2(n - \gamma)$ request vectors. We follow the same steps as above with n replaced by $n - \gamma$. For this, we will now need N to satisfy (cf. (13))

$$(n - \gamma)^2 - (n - \gamma) + 1 \leq N \leq 3(n - \gamma)^2 - (n - \gamma) + 1.$$

It is easy to verify that the left inequality follows from the definitions of n and γ . Hence, for $N \leq 3(n - \gamma)^2 - (n - \gamma) + 1$,

$$M + R \geq (n - \gamma) + \frac{N - ((n - \gamma)^2 - (n - \gamma) + 1)}{2(n - \gamma)}.$$

For $K < 2n$ and $N > 3(n - \gamma)^2 - (n - \gamma) + 1$, we proceed as before, but now the number of files N is larger than the number of indices u 's, v 's, and 1. We may set them all to be distinct files and hence, in step (iii), instead of decoding $N - ((n - \gamma)^2 - (n - \gamma) + 1)$ files, we now have $(3(n - \gamma)^2 - (n - \gamma) + 1) - ((n - \gamma)^2 - (n - \gamma) + 1)$ files. Thus,

$$\begin{aligned}
M + R & \geq (n - \gamma) + \frac{(3(n - \gamma)^2 - (n - \gamma) + 1) - ((n - \gamma)^2 - (n - \gamma) + 1)}{2(n - \gamma)} \\
& = 2(n - \gamma).
\end{aligned}$$

This completes the proof for $\alpha = 1$. For generalizing this to any $\alpha > 0$, we first consider the case of $N \geq \lceil \frac{1}{\alpha} \rceil$. For the case of $K \geq 2n$ (i.e., $\gamma = 0$), we consider $\lceil \frac{1}{\alpha} \rceil$ sets of $2n$ request vectors similar to (12). The condition analogous to (13) is now

$$\left\lceil \frac{1}{\alpha} \right\rceil (n^2 - n + 1) \leq N \leq \left\lceil \frac{1}{\alpha} \right\rceil (3n^2 - n + 1), \quad (15)$$

which can be verified to hold for n as defined in (5) with $\gamma = 0$. Now, in step (i), $\lceil \frac{1}{\alpha} \rceil$ files can be decoded by decoding one file from each of $\lceil \frac{1}{\alpha} \rceil$ sets of $2n$ request vectors. Then, in step (ii), we may now consider $2 \lceil \frac{1}{\alpha} \rceil$ sets

of n vectors each such that $n(n-1)$ files can be decoded from each set. The remaining $N - \lceil \frac{1}{\alpha} \rceil (n^2 - n + 1)$ can be decoded by combining all the $\lceil \frac{1}{\alpha} \rceil 2n$ vectors. Hence for $K \geq 2n$,

$$\begin{aligned} 2n \left(M + \left\lceil \frac{1}{\alpha} \right\rceil R \right) &\geq \left\lceil \frac{1}{\alpha} \right\rceil (2n) + \left\lceil \frac{1}{\alpha} \right\rceil (2n(n-1)) + N - \left\lceil \frac{1}{\alpha} \right\rceil (n^2 - n + 1) \\ &\geq 2 \left\lceil \frac{1}{\alpha} \right\rceil n^2 + N - \left\lceil \frac{1}{\alpha} \right\rceil (n^2 - n + 1). \end{aligned}$$

Since $\alpha M \geq \frac{M}{\lceil \frac{1}{\alpha} \rceil}$, we have, for $K \geq 2n$,

$$\alpha M + R \geq n + \frac{N - \lceil \frac{1}{\alpha} \rceil (n^2 - n + 1)}{2 \lceil \frac{1}{\alpha} \rceil n}. \quad (16)$$

The proof for $K < 2n$ is along the same lines as for $\alpha = 1$; as above, we now work with $\lceil \frac{1}{\alpha} \rceil 2(n - \gamma)$ request vectors instead of $\lceil \frac{1}{\alpha} \rceil 2n$.

When $N < \lceil \frac{1}{\alpha} \rceil$ we consider $\lceil \frac{1}{\alpha} \rceil$ request vectors such that one of the users, say the first user, requests all N files between these $\lceil \frac{1}{\alpha} \rceil$ request vectors. From this we get, $M + \lceil \frac{1}{\alpha} \rceil R \geq N$ which gives $\alpha M + R \geq \frac{N}{\lceil \frac{1}{\alpha} \rceil}$. This completes the proof for $\alpha > 0$.

Now we prove the second part of the Theorem 1 when $\alpha > 1$. We first consider the case of $K \geq 2 \lfloor \alpha \rfloor n$, where n is as defined in (9). Note that γ of (10) is 0 in this case. Now consider the following $2n$ request vectors.

$$(1, \dots, \lfloor \alpha \rfloor, u_1^1, \dots, u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^1, v_1^1, \dots, v_{n\lfloor \alpha \rfloor}^1, t_1^1, \dots, t_{K-2n\lfloor \alpha \rfloor}^1) \quad (17a)$$

$$(u_1^2, \dots, u_{\lfloor \alpha \rfloor}^2, 1, \dots, \lfloor \alpha \rfloor, u_{\lfloor \alpha \rfloor + 1}^2, \dots, u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^2, v_1^2, \dots, v_{n\lfloor \alpha \rfloor}^2, t_1^2, \dots, t_{K-2n\lfloor \alpha \rfloor}^2) \quad (17b)$$

\vdots

$$(u_1^n, \dots, u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^n, 1, \dots, \lfloor \alpha \rfloor, v_1^n, \dots, v_{n\lfloor \alpha \rfloor}^n, t_1^n, \dots, t_{K-2n\lfloor \alpha \rfloor}^n) \quad (17c)$$

$$(v_1^{n+1}, \dots, v_{n\lfloor \alpha \rfloor}^{n+1}, 1, \dots, \lfloor \alpha \rfloor, u_1^1, \dots, u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^1, t_1^{n+1}, \dots, t_{K-2n\lfloor \alpha \rfloor}^{n+1}) \quad (17d)$$

\vdots

$$(v_1^{2n}, \dots, v_{n\lfloor \alpha \rfloor}^{2n}, u_1^n, \dots, u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^n, t_1^{2n}, \dots, t_{K-2n\lfloor \alpha \rfloor}^{2n}) \quad (17e)$$

Of these, we require that $1, \dots, \lfloor \alpha \rfloor, u_1^1, \dots, u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^1, \dots, u_1^n, \dots, u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^n$ be distinct. Hence, we will require that $\lfloor \alpha \rfloor (n^2 - n + 1) \leq N$. Furthermore, we want these along with the v 's, i.e., $1, \dots, \lfloor \alpha \rfloor, u_1^1, \dots, u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^1, \dots, u_1^n, \dots, u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^n, v_1^1, \dots, v_{n\lfloor \alpha \rfloor}^1, \dots, v_1^{2n}, \dots, v_{n\lfloor \alpha \rfloor}^{2n}$ to include all of $1, 2, \dots, N$. Hence, we need n to be such that

$$\lfloor \alpha \rfloor (n^2 - n + 1) \leq N \leq \lfloor \alpha \rfloor (3n^2 - n + 1).$$

We can verify that the choice of n in (9), which is reproduced below, satisfies this.

$$n = \left\lceil \frac{\lfloor \alpha \rfloor + \sqrt{\lfloor \alpha \rfloor^2 + 12 \lfloor \alpha \rfloor (N - \lfloor \alpha \rfloor)}}{6 \lfloor \alpha \rfloor} \right\rceil.$$

Consider the first request vector and the first $n \lfloor \alpha \rfloor$ users. Users 1 to $\lfloor \alpha \rfloor$ request files W_1 to $W_{\lfloor \alpha \rfloor}$ and the rest $n \lfloor \alpha \rfloor - \lfloor \alpha \rfloor$ users request files $(W_{u_1^1}, \dots, W_{u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^1})$. Similarly in the second request vector, users $\lfloor \alpha \rfloor + 1$ to $2 \lfloor \alpha \rfloor$ request files W_1 to $W_{\lfloor \alpha \rfloor}$ and the rest $n \lfloor \alpha \rfloor - \lfloor \alpha \rfloor$ users request files $(W_{u_1^2}, \dots, W_{u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^2})$. This proceeds in the same manner until the n -th request vector. These $(W_1, \dots, W_{\lfloor \alpha \rfloor}, W_{u_1^1}, \dots, W_{u_{n\lfloor \alpha \rfloor - \lfloor \alpha \rfloor}^n})$ are $\lfloor \alpha \rfloor (n^2 - n + 1)$ distinct files in the database. For the second set of n request vectors, users $n \lfloor \alpha \rfloor + 1$ to $2n \lfloor \alpha \rfloor$ request the same files as users 1 to $n \lfloor \alpha \rfloor$ in the first n request vectors. For the first n request vectors, users $n \lfloor \alpha \rfloor + 1$ to $2n \lfloor \alpha \rfloor$ requests $n^2 \lfloor \alpha \rfloor$

files $(W_{v_1^1}, \dots, W_{v_{n[\alpha]}^n})$. For the second n request vectors, users 1 to n requests $n^2 \lfloor \alpha \rfloor$ files $(W_{v_1^{n+1}}, \dots, W_{v_{n[\alpha]}^{n+1}})$. By our choices we have ensured that these $2n^2 \lfloor \alpha \rfloor$ files contain the remaining $N - \lfloor \alpha \rfloor (n^2 - n + 1)$ distinct files.

We now follow the similar procedure as in the case when $0 < \alpha \leq 1$. First files W_1 to $W_{\lfloor \alpha \rfloor}$ can be decoded from all the $2n$ request vectors. This is done by considering the first request vector and caches Z_1 to $Z_{\lfloor \alpha \rfloor}$, the second request vector and caches $Z_{\lfloor \alpha \rfloor + 1}$ to $Z_{2\lfloor \alpha \rfloor}$ and so on for the remaining request vectors. Then, the first set of n vectors and the second set of n vectors are separately combined to decode files $(W_{u_1^1}, \dots, W_{u_{n[\alpha] - \lfloor \alpha \rfloor}^n})$. From the first n request vectors and caches $(Z_1, \dots, Z_{n[\alpha]})$ the files $(W_{u_1^1}, \dots, W_{u_{n[\alpha] - \lfloor \alpha \rfloor}^n})$ can be decoded. Similarly, from the second set of n vectors and $(Z_{n[\alpha] + 1}, \dots, Z_{2n[\alpha]})$ the same set of files can be decoded. The rest $N - \lfloor \alpha \rfloor (n^2 - n + 1)$ files which are included in $(W_{v_1^1}, \dots, W_{v_{n[\alpha]}^{2n}})$ can be decoded by considering all the $2n$ request vectors together using all the caches $(Z_1, \dots, Z_{2n[\alpha]})$. All R , M , entropies and mutual informations are in units of F bits. So for any achievable memory-rate pair (M, R) and $K \geq 2n \lfloor \alpha \rfloor$,

$$\begin{aligned} 2n(\lfloor \alpha \rfloor M + R) &\geq 2n \lfloor \alpha \rfloor + 2n(n \lfloor \alpha \rfloor - \lfloor \alpha \rfloor) + N - \lfloor \alpha \rfloor (n^2 - n + 1) \\ &\geq 2 \lfloor \alpha \rfloor n^2 + N - \lfloor \alpha \rfloor (n^2 - n + 1). \end{aligned}$$

Since $\alpha \geq \lfloor \alpha \rfloor$, for $K \geq 2n \lfloor \alpha \rfloor$,

$$\alpha M + R \geq n \lfloor \alpha \rfloor + \frac{N - \lfloor \alpha \rfloor (n^2 - n + 1)}{2n}.$$

The proof for $K < 2n \lfloor \alpha \rfloor$ is similar to the case of $0 < \alpha \leq 1$. Here we find the least integer γ such that $K \geq 2(n - \gamma) \lfloor \alpha \rfloor$. Notice that since $K \geq 2 \lfloor \alpha \rfloor$, $(n - \gamma) > 0$. Now we consider $2(n - \gamma)$ request vectors instead of $2n$. For this, we will now need N to satisfy

$$\lfloor \alpha \rfloor ((n - \gamma)^2 - (n - \gamma) + 1) \leq N \leq \lfloor \alpha \rfloor (3(n - \gamma)^2 - (n - \gamma) + 1).$$

It is easy to verify that the left inequality follows from the definitions of n and γ . Hence, for $N \leq \lfloor \alpha \rfloor (3(n - \gamma)^2 - (n - \gamma) + 1)$,

$$\alpha M + R \geq (n - \gamma) \lfloor \alpha \rfloor + \frac{N - \lfloor \alpha \rfloor ((n - \gamma)^2 - (n - \gamma) + 1)}{2(n - \gamma)}.$$

For $K < 2n \lfloor \alpha \rfloor$ and $N > \lfloor \alpha \rfloor (3(n - \gamma)^2 - (n - \gamma) + 1)$, we proceed as before, but now the number of files N is larger than the number of indices u 's, v 's, and $1, \dots, \lfloor \alpha \rfloor$. We may set them all to be distinct files and hence, in step (iii), instead of decoding $N - \lfloor \alpha \rfloor ((n - \gamma)^2 - (n - \gamma) + 1)$ files, we now have $\lfloor \alpha \rfloor (3(n - \gamma)^2 - (n - \gamma) + 1) - \lfloor \alpha \rfloor ((n - \gamma)^2 - (n - \gamma) + 1)$ files. Thus,

$$\begin{aligned} \alpha M + R &\geq (n - \gamma) \lfloor \alpha \rfloor + \frac{\lfloor \alpha \rfloor (3(n - \gamma)^2 - (n - \gamma) + 1) - \lfloor \alpha \rfloor ((n - \gamma)^2 - (n - \gamma) + 1)}{2(n - \gamma)} \\ &= 2(n - \gamma) \lfloor \alpha \rfloor. \end{aligned}$$

This completes the proof for $K < 2n \lfloor \alpha \rfloor$.

When $N < \lfloor \alpha \rfloor$ we consider $\lfloor \alpha \rfloor$ caches such that among them all K users are included. We consider one request vector where among the users all the N files are requested. Since $N < \lfloor \alpha \rfloor$ from the $\lfloor \alpha \rfloor$ caches all the files can be decoded, we get $\alpha M + R \geq N$. This completes the proof of Theorem 1 when $\alpha > 1$.

PROOF OF LEMMAS

Proof of Lemma 1.

Using equation (3), by substituting $N = \lceil \frac{1}{\alpha} \rceil K^2$ and $s = K$,

$$\begin{aligned} R^*(M) &\geq \left(K - \frac{K}{\lceil \frac{1}{\alpha} \rceil K^2 / K} M \right) \\ \frac{M}{\lceil \frac{1}{\alpha} \rceil} + R^*(M) &\geq K \\ \alpha M + R^*(M) &\geq K \end{aligned}$$

which gives,

$$N(\alpha, K) \leq \left\lceil \frac{1}{\alpha} \right\rceil K^2.$$

□

Proof of Lemma 2.

This proof follows from Theorem 1. Consider the case when K is even and $\alpha > 0$. We want to show that for

$$N = \left\lceil \frac{1}{\alpha} \right\rceil \left(\frac{3K^2}{4} - \frac{K}{2} + 1 \right), \quad (18)$$

the lower bound of Theorem 1 gives $\alpha M + R \geq K$. To see this, substitute N from (18) in (5)-(6) to see that $n = \frac{K}{2}$ and $\gamma = 0$. Then, the lower bound of (4) indeed gives $\alpha M + R \geq 2n = K$. Hence we have for even K ,

$$N(\alpha, K) \leq \left\lceil \frac{1}{\alpha} \right\rceil \left(\frac{3K^2}{4} - \frac{K}{2} + 1 \right).$$

To handle odd K as well, we note that $N(\alpha, K)$ is a non-decreasing function of K for fixed α . Hence for $\alpha > 0$ and $K \geq 2$,

$$N(\alpha, K) \leq \left\lceil \frac{1}{\alpha} \right\rceil \left(3 \left\lceil \frac{K}{2} \right\rceil^2 - \left\lceil \frac{K}{2} \right\rceil + 1 \right).$$

Following the same procedure for $\alpha > 1$ we first consider K to be such that $K = 2n \lfloor \alpha \rfloor$. We choose N to be,

$$N = \left(\frac{3K^2}{4 \lfloor \alpha \rfloor} - \frac{K}{2} + \lfloor \alpha \rfloor \right).$$

Then, the lower bound of (8) gives $\alpha M + R \geq 2n \lfloor \alpha \rfloor = K$. To find for any K , we note that $N(\alpha, K)$ is a non-decreasing function of K for fixed α . Hence for $\alpha > 1$ and $K \geq 2 \lfloor \alpha \rfloor$,

$$N(\alpha, K) \leq \lfloor \alpha \rfloor \left(3 \left\lceil \frac{K}{2 \lfloor \alpha \rfloor} \right\rceil^2 - \left\lceil \frac{K}{2 \lfloor \alpha \rfloor} \right\rceil + 1 \right).$$

Summarizing for $K \geq 2$ users and $\alpha > 0$,

$$N(\alpha, K) \leq \left\lceil \frac{1}{\alpha} \right\rceil \left(3 \left\lceil \frac{K}{2} \right\rceil^2 - \left\lceil \frac{K}{2} \right\rceil + 1 \right).$$

For $K \geq 2 \lfloor \alpha \rfloor$ users and $\alpha > 1$,

$$N(\alpha, K) \leq \lfloor \alpha \rfloor \left(3 \left\lceil \frac{K}{2 \lfloor \alpha \rfloor} \right\rceil^2 - \left\lceil \frac{K}{2 \lfloor \alpha \rfloor} \right\rceil + 1 \right).$$

□

Proof of Lemma 3.

To find the minimum number of files such that $(\alpha M + R_C(M))$ is K for the coded caching strategy explained in section III-B notice that,

$$\begin{aligned} \alpha M + R_C(M) &= \alpha M + \frac{K \left(1 - \frac{M}{N}\right)}{1 + \frac{KM}{N}} \\ &= \alpha M + \frac{KN - KM}{KM + N}. \end{aligned}$$

Since M takes only those values for which $\frac{MK}{N} \in \{1, 2, \dots, K\}$ as defined by the coded caching strategy we substitute $\frac{MK}{N} = 1$. Solving this we obtain,

$$N = \frac{1}{\alpha} \left(\frac{K^2}{2} + \frac{K}{2} \right). \quad (19)$$

To show that for all N less than (19), the scheme satisfies $\alpha M + R_C(M) < K$, consider $N = \left\lceil \frac{1}{\alpha} \left(\frac{K^2}{2} + \frac{K}{2} \right) - 1 \right\rceil$, $M = \frac{N}{K}$ and substitute in $\alpha M + R_C(M)$. We get,

$$\begin{aligned}
 \alpha M + R_C(M) &= \alpha M + \frac{KN - KM}{KM + N} \\
 &= \frac{\alpha N}{K} + \frac{K(1 - 1/K)}{2} \\
 &< \frac{\alpha(K^2 + K)}{2\alpha K} + \frac{K - 1}{2} \\
 &< \frac{K + 1}{2} + \frac{K - 1}{2} \\
 &< K.
 \end{aligned}$$

Hence,

$$N(\alpha, K) \geq \frac{1}{\alpha} \left(\frac{K^2}{2} + \frac{K}{2} \right).$$

□